# PERBANDINGAN KOMPLEKSITAS ALGORITMA A-STAR, FLOYD-WARSHALL, VITERBI PADA SDN (SOFTWARE DEFINED NETWORKING)

**Andi Tiana Putra[1], Drs. Ir. Rumani M., Bc.TT., M.Sc[2], Marisa W. Paryasto, S.T., M.T.[3]**

[1,2,]Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Univesitas Telkom
Jln. Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia
andi.id.pribadi@gmail.com[1], rumani@telkomuniversity.ac.id[2], marisa.paryasto@gmail.com[3]

## ABSTRAK

Algoritma Pencarian jalur terpendek atau lebih dikenal sebagai *shorterst-path* dipakai dalam menentukan rute dalam sebuah graff. Ini dilakukan untuk menentukan jalur terpendek dari titik awal sampai titik tujuan dalam sebuah graff.Algoritma pencarian jalur terpendek sendiri sudah diimplementasikan didalam jaringan, ini dilakukan untuk mencari jalur atau rute terpendek dari suatu host awal ke host tujuan dengan topologi tertentu. Dengan adanya banyak algoritma pencarian jalur terpendek tentunya memberikan kita pilihan untuk menentukan algoritma mana yang ingin digunakan. Disamping itu kita harus menentukan algoritma manakah yang memiliki waktu komputasi tercepat dan penggunaan ruang memori yang efisien.

Dalam penelitian ini dilakukan perbandingan kompleksitas algoritma pada jaringan SDN. dimana tujuannya adalah menentukan algoritma yang paling baik dalam sebuah topologi dengan menentukan kompleksitas masing masing algoritma. Dalam penelitian ini dilakukan pengujian QoS dengan parameter *delay*, *delay* yang dimaksud adalah *one way delay* (*latency*) dan *resource utilization.* juga mininet dipilih sebagai emulator jaringan SDN yang yang bertindak sebagai *data plane*, sementara RYU dipilih sebagai controller yang bertindak sebagai *control plane*. Dan protokol yang dipilih adalah protokol OpenFlow. Penggunaan algoritma A-star, Floyd-warshall, dan Viterbi sebagai penjaluran yang diimplentasikan didalam *controller*. dengan topologi yang digunakan yaitu *mesh* dan *tree* dengan penggunaan variasi switch yang berbeda yakni 5,8, dan 11.

Berdasarkan pengujian didapat nilai *delay* dan *resource utilites* dengan mengacu pada standarisasi ITU.T G.1010 dengan nilai rata rata *delay* < 60s dan *packet loss* 0%. sehingga dapat dikatakan layanan ini sudah baik menurut standarisasi ITU.T G.1010. Didapatkan juga bahwa kompleksitas total yang didapatkan dari pengujian menunjukan bahwa algoritma Floyd-warshall memiliki nilai kompleksitas paling rendah pada topologi mesh dan algoritma viterbi memiliki kompleksitas paling rendah pada topologi tree..

**Kata Kunci**: *Kompleksitas, SDN, Algoritma, Ryu*

## ABSTRACT

The shortest path search algorithm known as the shorterst-path is used in determining the route in a graff. This is done to determine the shortest path from the starting point to the destination point in a graff. The shortest path search algorithm itself has been implemented in the network, this is done to find the path or shortest route from an initial host to a destination host with a certain topology. With many shortest path search algorithm give us the option to determine which algorithm want to use. Besides, we must determine which algorithm has the fastest computing time and efficient use of memory space. In this research we do comparison of algorithm complexity in SDN network. where the goal is to determine the best algorithm in a topology by determining the complexity of each algorithm.

In this research is QoS testing with delay parameter, delay is one way delay (latency) and resource utilization also mininet is selected as an SDN network emulator as data plane, while RYU is selected as a control plane. And the protocol chosen is the OpenFlow protocol. The use of the A-star, Floyd-warshall, and Viterbi algorithms as routing is implemented within the controller. with the topology used is the mesh and tree with the use of different switch variations 5.8, and 11.

As result the test obtained the value of delay and resource utilites with reference to ITU.T. G.1010 standardization with the average value of delay <60s and packet loss 0%. so it can be said that this service is good according to ITU.TT G.1010. It was also found that the total complexity obtained from the tests shows that the Floyd-warshall algorithm has the lowest complexity value in mesh topology and Viterbi algorithm has the lowest complexity value in tree topology..

**Keywords***: Complexity, SDN, Algorithm, Ryu*

## I. PENDAHULUAN

### 1.1. Latar Belakang

A Algoritma pencarian jalur terpendek atau lebih dikenal sebagai *shortest-path* dipakai dalam menentukan rute dalam sebuah graff. Ini digunakan untuk menentukan jalur terpendek dari titik awal ke titik tujuan dalam sebuah graff. Algoritma pencarian jalur terpendek sendiri sudah diimplementasikan didalam berbagai hal, salah satunya didalam jaringan. Perkembangan jaringan komputer yang semakin besar mengakibatkan dalam jaringan komputer sering kali antara satu komputer dengan komputer lain terpisah jauh secara geografis, Hal ini menimbulkan kebutuhan yang tinggi terhadap performansi jaringan dan control jaringan yang semakin rumit dan tidak fleksibel. Kinerja suatu jaringan dapat dipengaruhi oleh beberapa faktor seperti konfigurasi jaringan, jumlah permintaan, dan metode manajemen jaringan tersebut. Salah satu yang faktor yang mempengaruhi manajemen jaringan adalah network routing[6].

Routing merupakan penentuan rute terbaik yang akan dilalui informasi yang dikirim dari pengirim menuju penerima[6]. Dalam network, routing biasanya menggunakan suatu algoritma pencarian jalur terpendek. Dengan adanya banyak algoritma pencarian jalur terpendek atau *shortest-path* tentunya memberikan kita banyak pilihan dalam menentukan algorima yang ingin kita gunakan. Disamping itu kita harus menetukan algoritma manakah yang memiliki waktu komputasi paling cepat dan seberapa besar penggunaan ruang memori ketika komputasi

berlangsung. disinilah kompleksitas digunakan. kompleksitas algoritma diukur berdasarkan kinerjanya dengan menghitung waktu eksekusi suatu algoritma[9]. waktu eksekusi algoritma dapat diklasifikasikan menjadi tiga kelompok besar, yaitu *best-case* (kasus terbaik), *average-case* (kasus rata-rata) dan *worst-case* (kasus terburuk)[1].

Menanggapi dari latar belakang diatas mendorong penulis untuk melakukan penelitian perbandingan kompleksitas antara algoritma A*(A star),Floyd-Warshall, dan *Viterbi* pada dua topologi berbeda yaitu *Mesh* dan *Tree* didalam jaringan SDN. dan akan dilakukan analisis kinerja dari masing masing algoritma pada topologi *mesh* dan *Tree* untuk dimenentukan algoritma terbaik untuk masing masing topologi tersebut.

## II. LANDASAN TEORI

### 2.1. Kompleksitas Algoritma

Kompleksitas dibagi menjadi dua jenis yaitu kompleksitas waktu (*time complexity*) dan kompleksitas ruang (*space complexity*). Kompleksitas waktu atau *time complexity* adalah jumlah intruksi yang dijalankan suatu program selama waktu berjalannya. Kompleksitas waktu diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan suatu algoritma sebagai fungsi masukan *n* [1]. Kompleksitas waktu dibedakan menjadi tiga yakni kompleksitas waktu terbaik (*best case*), kompleksitas waktu rata-rata (*average case*), dan kompleksitas waktu terburuk (*worst case*). Kompleksitas ruang atau *space complexity* bisa didefinisikan sebagai pengukuran jumlah ruang atau memori yang digunakan algoritma untuk dijalankan sebagai fungsi dari panjang suatu input[9]. Kompleksitas memori atau area dapat dilihat dari seberapa besar memori yang dipakai ketika program sedang dieksekusi. Sendangkan kompleksitas total adalah hasil kali kompleksitas area(memori) dengan hasil kuadrat kompleksitas waktu [8].

### 2.2. Algoritma A-star

Algoritma A Star atau A* adalah salah satu algoritma pencarian yang menganalisa input, mengevaluasi sejumlah jalur yang mungkin dilewati dan menghasilkan solusi. Algoritma A* adalah algoritma komputer yang digunakan secara luas dalam graph traversal dan penemuan jalur serta proses perencanaan jalur yang bisa dilewati secara efisien di sekitar titik-titik yang disebut node [4]. Karakteristik yang menjelaskan algoritma A* adalah pengembangan dari "daftar tertutup" untuk merekam area yang dievaluasi. Daftar tertutup ini adalah sebuah daftar untuk merekam area berdekatan yang sudah dievaluasi, kemudian melakukan perhitungan jarak yang dikunjungi dari "titik awal" dengan jarak diperkirakan ke "titik tujuan" [4].

Algoritma A* menggunakan path dengan *cost* paling rendah dari titik awal ke titik tujuan. Secara matematika A* menggunakan rumus

$$f(x) = g(x) + h(x)$$

Dimana :

*f(x)* adalah fungsi perkiraan saat ini dari jarak terdekat ke suatu tujuan.

*g(x)* adalah jarak total dari posisi awal ke posisi sekarang.

*h(x)* adalah fungsi heuristik yang digunakan untuk meperkirakan jarak dari suatu *node* ke *node* tujuan.

### 2.3. Floyd-Warshall

Algoritma Floyd-warshall adalah algoritma untuk menemukan jalur terpendek dalam *weighted graph*. Algoritma ini akan membandingkan semua jalur yang mungkin melalui grafik antara masing masing simpul (*edge*).

Anggap ada sebuah graf G dengan titik V dengan urutan 1 sampai N. dimana notasi $d_{ijk}$ yang berarti jalur terpendek dari *i* ke *j* , yang juga melewati titik *k*. tentunya jika memang ada jalur antara titik *i* dan *j* maka notasi akan menjadi $d_{ij0}$. Bagaimanapun nilai dari $d_{ijk}$ mempunyai dua pilihan: pertama jika jalur terpendek dari *i* ke *j* tidak melalui titik *k* maka nilai $d_{ijk}$ akan sama dengan $d_{ijk-1}$. Kedua jika jalur terpendek dari *i* ke *j* melewati titik *k* dimana pertama tama dimulai dari *i* menuju *k* setelah itu pergi dari *k* meneuju *j*. dalam kasus ini nilai dari $d_{ijk}$ akan sama dengan $d_{ikk-1} + d_{kjk+1}$[7]. Dan untuk menentukan jalur terpendek kita hanya perlu menemukan nilai minimum dari dua pernyataan ini :

$d_{ij0}$= panjang jalur antara titik *i* dan *j*.

$d_{ijk} = \min(d_{ijk-1}, d_{ikk-1} + d_{kjk-1})$

### 2.4. Viterbi

Algoritma Viterbi adalah algoritma pemgrograman dinamis untuk menemukan urutan yang paling mungkin dari *hidden state* yang disebut jalur Viterbi yang menghasilkan urutan kejadian yang teramati, terutama dalam konteks *Markov information source* dan model *hidden Markov*[3].

Pada algortma Viterbi untuk penyelesaian dinamis program Markov model pada urutan pengamatan sepanjang *n* memiliki aturan sebalagi berikut :

1. Inisialisasi $\delta_0(s) = 1$ dimana *s* adalah state awal, dan $\delta_0(s) = 0$ untuk state lainnya.
2. Setiap nilai *i = 1, . . . . , n,* hitung :

$$\delta_i(s) = \max_{s_{i-1}} P(s_i \mid s_{i-1}) P(w_{i-1} \mid s_{i-1}) \delta_{i-1}(s_{i-1})$$

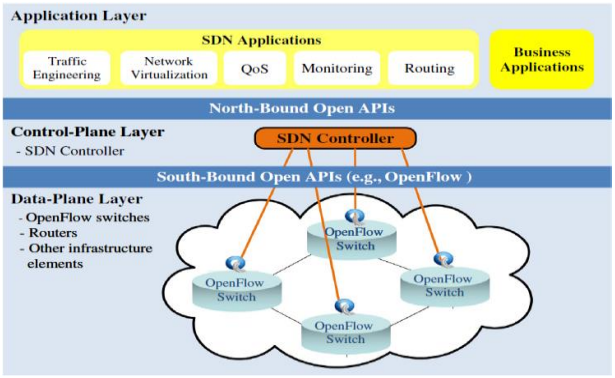
$$\psi_i(s) = \arg\max_{s_{i-1}} P(s_i \mid s_{i-1}) P(w_{i-1} \mid s_{i-1}) \delta_{i-1}(s_{i-1})$$

3. Kemudian isi *end state* (posisi *n + 1*) menggunakan aturan pada (2) diatas.

### 2.5. SDN (*Software Defined Networking*)

*Sotware Defined Networking* (SDN) adalah sebuah konsep pada jaringan dimana perangkat *control plane* dan *data plane* dibuat terpisah, berbeda dengan jaringan tradisional dimana perangkat *control plane* dan *data plane* masih menjadi satu[5]. Pemisahan ini memungkinan seseorang untuk menerapkan *control plane* di suatu entitas eksternal, yang disebut *controller* dan *data plane* yang termasuk pada *forwarding*

*packet* disebut *switch*[6]. Konsep SDN yang mempunyai arsitektur yang memisahkan fungsi *control plane* dan *forwarding packet* yang memungkinkan kontrol jaringan menjadi *programmable* dan infrastruktur yang mendasarinya akan diabstraksikan untuk aplikasi dan layanan jaringan. SDN arsitektur adalah sebagai berikut :

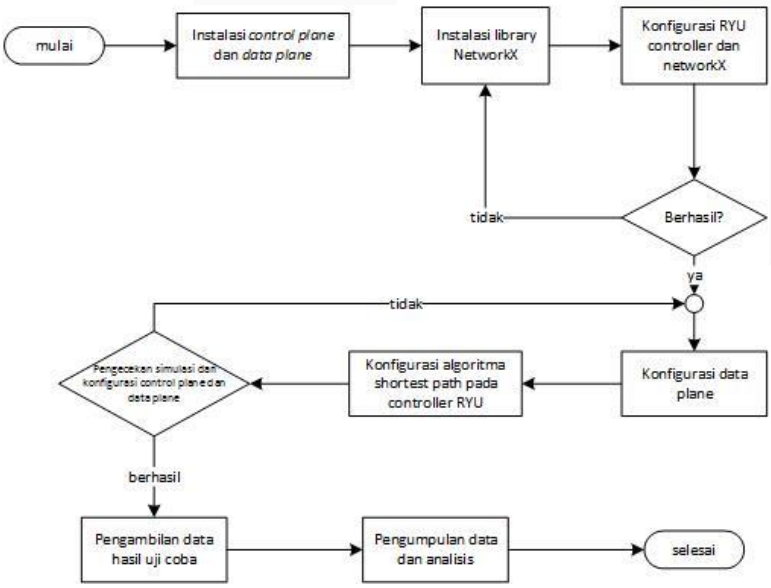



*gambar 2.1 SDN arsitektur [6]*

## III. PERANCANGAN SISTEM

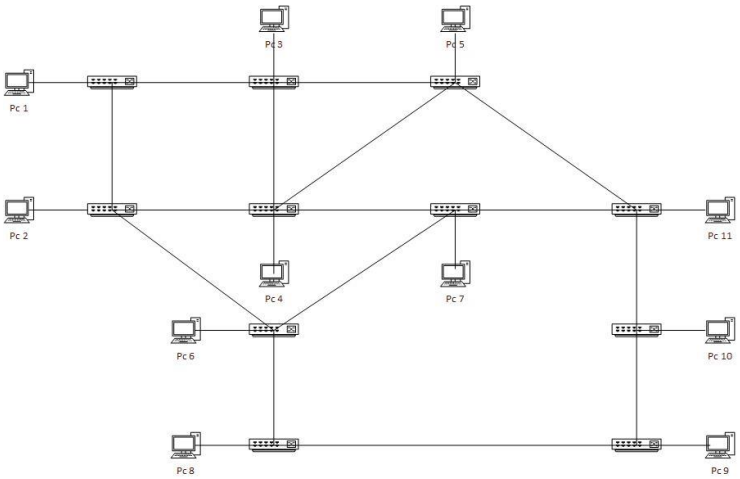
### 3.1. Skema Sistem

Pada Perancangan sistem pertama-tama dilakukan penentuan sistem untuk melakukan simulasi, yakni dengan menginstall beberapa aplikasi seperti mininet , ryu controller, dan library networkX kemudian dilakukan pengecekan setelah installasi berhasil. Selanjutnya dilakukan perancangan topologi jaringan yang dipakai dalam penelitian ini yaitu topologi mesh dan topologi tree didalam mininet dengan variasi switch yang berbeda yaitu 5, 8, dan 11 switch. Selanjutnya membuat API untuk RYU controller yang menganding routing dimana didalam API ini terdapat Algoritma yang dipakai dalam penelitian yakni algoritma A-star, Floyd-warshall, dan Viterbi sebagai penentu jalur. Setelah itu dilakukan pengecekan pada kedua rancangan yaitu mininet dana ryu controller dengan menghubungkan mininet dengan ryu controller kemudian melakukan perintah pingall pada terminal. Selanjutnya adalah pengujian dengan skema dan parameter uji yang ditentukan. Dan proses pengambilan data hasil dari pengujian sebanyak 30 kali dari masing masing topologi dan variasi jumlah switch. Kemudian menganalisis data yang diperoleh dari hasil pengujian yang akan direpresentasikan dalam bentuk grafik dan tabel.

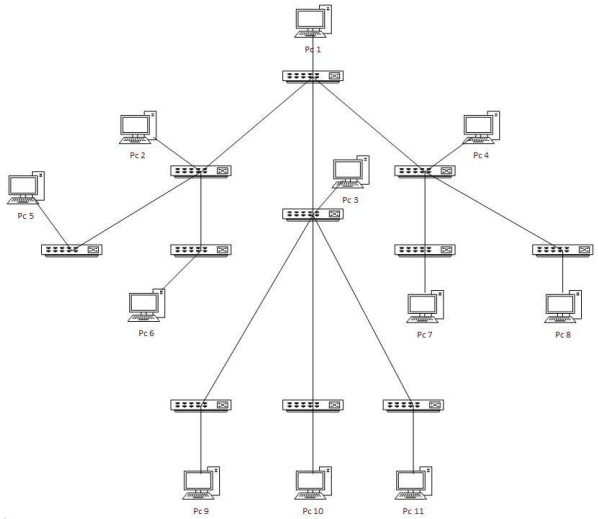



*gambar 3.1 Skema Sistem*

### 3.2. Perancangan Topologi




*gambar 3.2 Topologi Mesh*




*gambar 1.3 Topologi Switch*

Dalam perancagan penelitian ini dirancang topologi mesh dan tree dengan variasi switch 5, 8, dan 11. dimana pada penelitian kali ini satu buat switch memliki satu buah host. Adapun spesifikasi yang dipakai dalam pembuatan topologi yang dijelaskan pada tabel 3.1.

| Topologi | Jumlah switch | Jumlah host | Jumlah link antar switch | Bandwith link | Delay link | Metric |
|---|---|---|---|---|---|---|
| Mesh | 5 | 5 | 6 | 70 Mpbs – 100 Mpbs | 1 – 7 ms | (bandwith + delay)*256 |
| | 8 | 6 | 11 | | | |
| | 11 | 11 | 16 | | | |
| Tree | 5 | 5 | 4 | | | |
| | 8 | 8 | 7 | | | |
| | 11 | 11 | 10 | | | |

*Tabel 3.1 spesifikasi switch*

### 3.3. Perancangan RYU *Controller*

Pada perancangan RYU yang bertindak sebagai *control plane* yang bertugas sebagai kontroler yang menggunakan protokol OpenFlow. dikarenakan protokol OpenFlow mampu membuat *control plane* berinteraksi dengan *data plane* (switch) dengan menggunakan Open Vswitch sebagai switch pada layer 2. Ryu memiliki API, API ini bertujuan untuk memudahkan dikembangkan dengan manajemen dan kontrol jaringan yang baru. Pada API kali ini dibuat API yang mengandung algoritma A-star, Floyd-warshall, dan Viterbi.

## IV. IMPLEMENTASI DAN PENGUJIAN

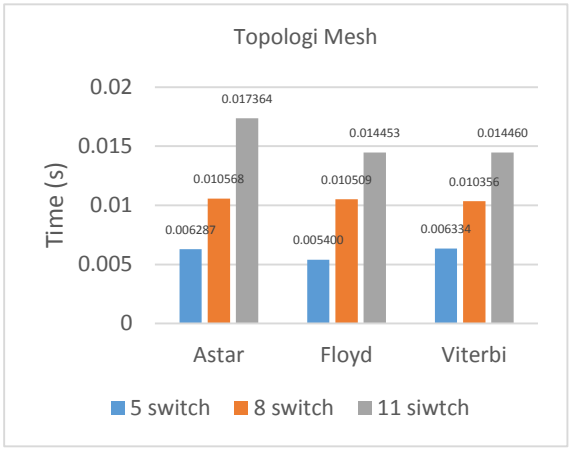
### 4.1. Pengujian QoS

Pada pengujian QoS, dipilih parameter delay sebagai acuan untuk menentukan waktu yang dibutuhkan algoritma. dengan skenario mengirim data yang memiliki karakteristik pada layanan UMTS. Maka dipilih paket informasi berupa WWW, pada transport layer UDP dengan rata-rata *time-arrival* 0.125 yang membutuhkan minimal 32 Kbps.

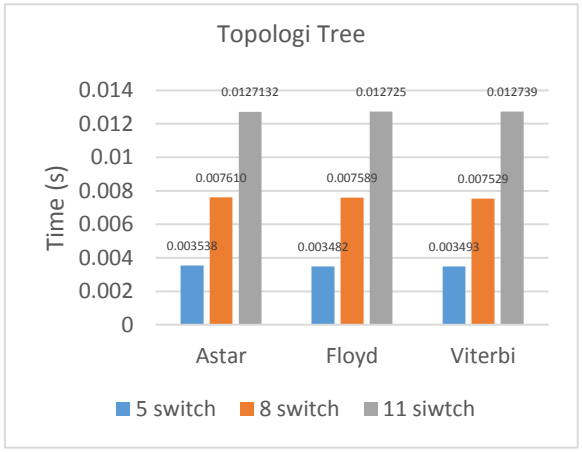
| Packet based information types | Average number of packet calls within a session | Average reading time between packet calls | Average amount of packet within a packet call | Average inter arrival time between packets (s) | Parameters for packet size distribution (pareto distribution) |
|---|---|---|---|---|---|
| WWW sufing UDD 8Kbit/s | 5 | 412 | 25 | 0.5 | k=81.5 $\alpha$ = 1.1 |
| WWW sufing UDD 32Kbit/s | 5 | 412 | 25 | 0.125 | k=81.5 $\alpha$ = 1.1 |
| WWW sufing UDD 64Kbit/s | 5 | 412 | 25 | 0.0625 | k=81.5 $\alpha$ = 1.1 |
| WWW sufing UDD 144Kbit/s | 5 | 412 | 25 | 0.0277 | k=81.5 $\alpha$ = 1.1 |
| WWW sufing UDD 384Kbit/s | 5 | 412 | 25 | 0.0104 | k=81.5 $\alpha$ = 1.1 |
| WWW sufing UDD 2048Kbit/s | 5 | 412 | 25 | 0.00195 | k=81.5 $\alpha$ = 1.1 |

*Tabel 4.1 paket UMTS [8]*

Dengan hasil yang diperoleh dari pengujian masing masing topologi sebagai berikut :




*gambar 4.1 grafik pengujian waktu pada topologi mesh*

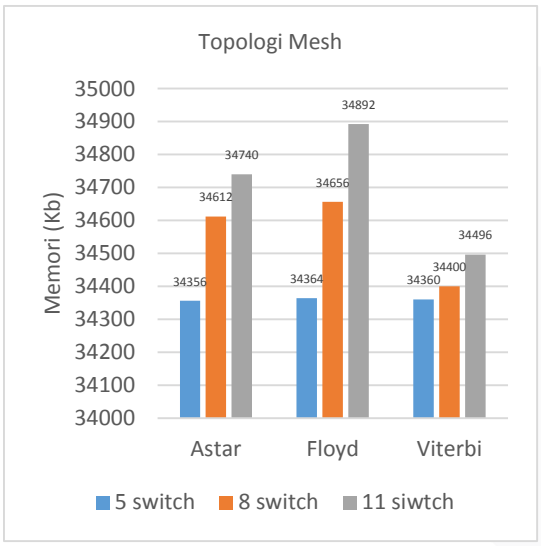



*gambar 4.2 grafik pengujian waktu pada topologi tree*

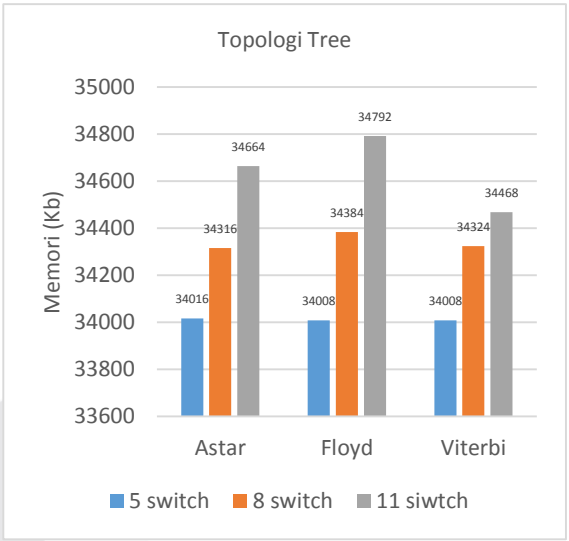
Dari gambar 4.1 dan 4.2 dapat diambil kesimpulan bahwa seiring dengan penambahan switch waktu yang dibutuhkan juga meningkat.

### 4.2. Pengujian *Resource Utilization*

Pengujian *resource utilization* dilakukan guna mengetahui memori yang dipakai ketika program sedang dieksekusi. Untuk melihat berapa memori yang dipakai, digunakan *command top* pada Ubuntu untuk melihat memori. Dan hasil yang didapat adalah sebagi berikut :




*gambar 4.3 grafik pengujian memori pada topologi mesh*




*gambar 4.4 grafik pengujian waktu pada topologi tree*

Dapat disimpulkan dari 4.3 dan 4.4 memori meningkat seuring dengan penambahan switch. Dan algoritma yang mengkonsumi memori paling kecil sampai 11 switch adalah algoritma Viterbi.

### 4.3. Analisis Kompleksitas

Nilai kompleksitas total didapat dari pengujian sebelumnya, dimana pengambil nilai kompleksitas total adalah hasil perkalian dari kompleksitas area dengan hasil kuadrat kompleksitas waktu[marisa]. Maka diperoleh kompleksitas total dari masing masing topologi sebagai berikut :

| Kompleksitas Total Topologi Mesh (AT^2) | | | |
|---|---|---|---|
| | Astar | Floyd | Viterbi |
| 5 switch | 1.357967933 | 0.142324011 | 1.378507944 |
| 8 switch | 3.865558982 | 1.980274996 | 3.689287718 |
| 11 siwtch | 10.47440515 | 5.633716345 | 7.212823834 |

*tabel 4.2 hasil pengujian kompleksitas total topologi mesh*

| Kompleksitas Total Topologi Tree (AT^2) | | | |
|---|---|---|---|
| | Astar | Floyd | Viterbi |
| 5 switch | 0.425793375 | 1.00205424 | 0.414933274 |
| 8 switch | 1.987311624 | 3.827376791 | 1.945684806 |
| 11 siwtch | 5.602584746 | 7.28856228 | 5.593540147 |

*Tabel 4.3 hasil pengujian kompleksitas total topologi tree*

dari tabel 4.2 diketahui bahwa algoritma Floyd-warshall memiliki nilai kompleksitas paling kecil pada topologi mesh yang memiliki jumlah switch hingga 11 switch, sehingga dapat dikatakan bahwa algoritma Floyd-warhsall adalah algoritma yang paling baik diterapkan dalam topologi mesh. Sementara itu dari tabel 4.3 dapat diketahui bahwa algoritma Viterbi memiliki nilai kompleksitas total paling kecil disbanding kedua algoritma lainnya.

## V. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Kesimpulan yang didapat dari penelitian ini adalah seiring bertambahnya jumlah switch kebutuhan waktu dan penggunaan memori juga meningkat. Begitu juga dengan kompleksitas total dari algoritma yang mana semakin kecil nilai dari kompleksitas total maka semakin baik pula algoritma tersebut. Diketahui bahwa algoritma Floyd-warshall memiliki nilai kompleksitas terkecil pada topologi mesh yang menggunakan switch hingga 11 switch. Sehingga ini membuktikan bahwa algoritma Floyd-warshall adalah algoritma yang paling baik yang dapat diimplementasikan pada topologi mesh. Sementara pada topologi tree yang menggunakan switch hingga 11 switch bahwa algoritma Viterbi memiliki nilai kompleksitas paling kecil yang membuktikan algoritma Viterbi adalah algoritma yang paling baik untuk diimplementasikan dalam topologi tree ketimbang dua algoritma lainnya.

### 5.2. Saran

Begitupun ada saran dari penulis yakni, penggunaan jumlah switch yang lebih banyak dari penelitian ini, dan dengan keadaan traffic yang sibuk apakah hasilnya berubah ataukah hasilnya akan sama dengan penelitian ini.

## Daftar Pustaka

[1] Azizah, U. N. (2013). Perbandingan Detekrok Tepi Prewit dan Detekror Tepi Laplacian Berdasarkan Kompleksitas Waktu dan Citra Hasil.

[2] Dibi Khairurrazi Budiarsyah. (2013/22014). Analisis Kompleksitas Waktu Untuk Beberapa Algoritma Pengurutan. *Makalah IF2120 Matematika Diskrit*.

[3] G. DAVID FORNEY, J. (1973). The Viterbi Algorithm. *PROCEEDINGS OF THE IEEE, VOL. 61, NO. 3,*.

[4] H, R. (2013). Path Finding-Djikstra's and A* Algorithm.

[5] Ian F. Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, Wu Chou. (2014). A roadmap for traffic engineering in software defined networks.

[6] Ilhamsyah, M. (2016). PERANCANGAN SDN (SOFTWARE DEFINED NETWORK) DENGAN ALGORITMA JOHNSON MENGGUNAKAN EMULATOR MININET.

[7] Kairanbay Magzhan, Hajar Mat Jani. (2013). A Review And Evaluations Of Shortest Path Algorithms. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 6,*.

[8] PARYASTO, M. W. (2012). ARSITEKTUR UNIT PENGALI COMPOSITE FIELD KOMBINASI MH-KOA UNTUK ELLIPTIC CURVE CRYPTOGRAPHY.

[9] Subandijo. (2011). EFISIENSI ALGORITMA DAN NOTASI O-BESAR.

[10] *Universal Mobile Telecommunications System (UMTS) Selection procedures for the choice of radio transmission technologies of UMTS (UMTS 30.03 version 3.2.0).* (2004). European Telecommunications Standards Institute.